```
(require spd/tags)

(@htdd Trip)
(define-struct trip (origin destination mot duration))
;; Trip is (make-trip String String String Natural)
;; interp. a trip with origin, destination, mode of transport,
;;          and duration in days

(define T1 (make-trip "Vancouver" "Cancun" "Flight" 10))
(define T2 (make-trip "Calgary" "Ottawa" "Car" 14))
(define T3 (make-trip "Montreal" "New York" "Flight" 5))

(@dd-template-rules compound) ;4 fields


(define (fn-for-trip t)
  (... (trip-origin t)        ;String
       (trip-destination t)  ;String
       (trip-mot t)          ;String
       (trip-duration t)))   ;Natural


(@htdf longer-trip)
(@signature Trip Trip -> Trip)
;; produce trip with longer duration, if durations are equal produce 2nd trip
(check-expect (longer-trip T2 T3) T2)
(check-expect (longer-trip T3 T1) T1)
(check-expect (longer-trip T3 (make-trip "Houston" "Dallas" "Car" 5))
              (make-trip "Houston" "Dallas" "Car" 5))

;(define (longer-trip t1 t2) T1)  ; stub

(@template-origin Trip)

(@template
 (define (fn-for-trip t1 t2)
   (... (trip-origin t1)
        (trip-destination t1)
        (trip-mot t1)
        (trip-duration t1)
        (trip-origin t2)
        (trip-destination t2)
        (trip-mot t2)
        (trip-duration t2))))

(define (longer-trip t1 t2)
  (if (> (trip-duration t1) (trip-duration t2))
      t1
      t2))

#|
[A] There is a question expression on line 47.

[B] The true answer in the function body of longer-trip is a call to >.

[C] The function body spans lines 36 to 44.

[D] Line 47 contains a call to a primitive function.
```

```
59
60    [E] Line 4 is a type comment.
61
62    [F] Trip is compound data with 4 atomic distinct fields.
63
64    [G] There is an if statement in the function body of longer-trip.
65
66    [H] Line 40 contains a call to a selector.
67
68    [I] In the function definition for longer-trip, the arguments
69        are t1 and t2.
70    |#
```